Published in: *ACM Sigplan Notices*, June 2001, pg.23-25.
Updated: July 21, 2015

Forth Report

Editor: *Paul Frenger, P.O. Box 820506, Houston, TX 77282; pfrenger@ix.netcom.com*

Is Forth Dead? Updated!

Paul Frenger

1 Another Dead Language?

My wife's father was a food chemist who pioneered a technique for putting Hawaiian pineapples into tin cans. He had a semiclassical education. When she was a little girl, he taught her the following little ditty:

Latin is a dead language
As dead as it can be.
It killed all the Romans
And now it's killing me.

Recent events may suggest to some that substituting the word "Forth" for "Latin" and "embedded programmers" for "Romans" would yield an even more truthful stanza. Forth, you will recall, is the language created by Chuck Moore: a dual-stack, reverse Polish, interpreter / incremental compiler originally designed for resource constrained systems, and often achieved as a standalone *language-cum-operating system.*

The first news item is the collapse of the U.S. Forth Interest Group, and its flagship publication *Forth Dimensions.* The second item is the fact that the Forth-based IEEE standard 1275 ("Open Firmware") will not be renewed. These events and others will be discussed shortly.

Many of the pioneers of Forth date back to its earliest days in the 1970's. Their hair is gray or white (or worse ... vanished). Some of them have left the fold; others have gone to retirement or to rest. Fine old meetings like the "Rochester Forth Conference" and "ACM SIGForth Workshop" have entirely disappeared. Who if anyone is picking up the torch for them to carry to new generations of programmers?

2 FIG: Requiescat in Pace?

The Forth Interest Group (better known as "FIG") has been around for over twenty years [1]. It emerged from the desire of Bill Ragsdale and his friends (Kim Harris, John James, David Boulton, Dave Bengel, Tom Olsen, and Dave Wyland, among others) to devise a low cost version of Forth for every microprocessor then available. To this end they succeeded; their work product was called *FIG-Forth* and it is still available today [2]. FIG was a noble and mighty undertaking which effectively spread the word to engineers and programmers around the world; the role of FIG was elucidated and praised in these pages not too long ago [3].

Then there is the bad news. In February, 2000, FIG stopped accepting membership renewals or orders for items it sold through its mail-order store. A management shakeup took place, resulting in the departure of the existing FIG officers. John Hall took over as acting president of FIG and continued maintenance of its website [4]. Calling the FIG office [5] in Lafayette, California, one gets a recording made by John which states that FIG's offices are closed and that publication of *Forth Dimensions* has been suspended ... possibly permanently. He says that FIG is in a "slow restart" mode, whatever that means. We have to accept the possibility that FIG may never recover from this disaster.

3 Quo Vadis, IEEE1275?

IEEE standard 1275 (aka: "Open Firmware") describes a machine independent system for booting up personal computers, workstations and networks. This software system was described in general terms in this journal two years ago [6] and more specifically elsewhere [7]. Open Firmware has been used in Sun workstations since 1986, all Apple Macs since 1995 and recent IBMs and possibly other machines. It is a very sophisticated "plug-and-play" mechanism which starts up a computer, finds / initializes all the devices attached to it, then loads the native operating system and "disappears". The crux of IEEE1275 is that programs are written in a tokenized Forth source code which is stored in ROM on the various system boards, compiled on startup and executed. For the IBM PC, clever address mapping allows a board (e.g. a video card) to have both DOS compatible and IEEE1275 compatible startup code in the same ROM without contention.

Again, now for the bad news. In a `comp.lang.forth` posting [8], a statement by Mitch Bradley (initial developer of IEEE1275) is quoted which indicates that the industry consortium (Sun, Apple, IBM, etc) which pushed through the initial standard was not going to renew it. Creating a standard such as "Open Firmware" is very expensive and time consuming, and the standard must periodically be revisited or it will lapse. Mitch says that the people who put IEEE1275 together can work together very well, thank you, without any external "standard" process to bother with. Well, if you say so, but letting standardized software systems "evaporate" makes me quite nervous.

4 Et Tu, ANS Forth?

Worse still, a "zinger" was contained in that posting: Elizabeth Rather (President of Forth, Inc., the oldest commercial supplier of Forth systems) mentioned that the Forth vendors who were responsible for the ANS Forth standard were also exhausted, and didn't think they would be able to justify the work and expense necessary to continue ANS Forth. This may mean that yet another Forth-based standard will be biting the dust in the near future.

Why get all excited about these changes? Well, at the time of their inception (especially ANS Forth), we in the Forth community were told that having a well-known standard would catapult Forth into the consciousness of every programmer, thereby increasing its usage as its status rose.

Didn't happen. Most programmers are stuck with C or C++ or Java and THAT IS IT. No leeway, no option. Besides, "real" programmers consider Forth to be an engineer's language, like "ladder logic" and CNC lathe control languages. No programmer sees any status in these "grunt" languages.

The previous Forth standard was *Forth-83*, which was created by an ad hoc "Forth Standards Team" and promulgated through FIG. Forth-83 was not as elegant as ANS Forth, but it was easy on hardware. But now you can see the problem: no FIG, no ANS Forth, no new Forth standard ... no new Forth programmers.

5 Episode Four: A New Hope

Is there any reason for expecting that Forth may yet persist as a viable programming language for awhile longer? Yes, possibly; please consider the following.

The circulation of *Forth Dimensions* and *ACM SIGForth* were never very high: when I was editor of the latter, we printed just over 1000 copies per issue at its peak distribution. Many of these were sent to libraries, not individuals. It is likely that more programmers read the manuals supplied with Forth software than any magazine which spotlights Forth applications. It is a well known fact that Forth programmers in general won't contribute articles to Forth publications; less well known is the likelihood that they do not read such publications, either.

Although the death of FIG USA (note the suffix) appears to be "res ipsa loquitur" now, there is more to FIG than just California. For example, FIG UK is alive and well; we recently published a guest column from one of their members [9] who is editor of their *Forthwrite Magazine*. Then there is the German FIG, and the Dutch FIG, and FIGs all over the world. It is hard to accept that Forth, a language born in the good old USA, is considered obsolete and shunned here. In Europe, for example, Forth is very popular and is used for serious computer systems [10] such as automatic teller machines (ATMs) and control of a portion of the London subway system. The EuroForth Conferences are alive and well (I have enjoyed three, myself, so far). Our overseas programmer friends are not so persnickety about being found in the company of a Forth compiler as are we, ourselves. Too bad; it is our loss.

At least, NASA still loves Forth [11]. NASA uses Forth microprocessors (such as the Harris, now Intersil, RTX-2010) and the Forth language for such momentous projects as the "NEAR" mission. Space engineers like Forth's easy to use debugging features, interpreter / compiler interface, virtual mass storage and small memory footprint, among other things.

For the paranoid among you, I cannot resist mentioning that in addition to Forth underlying the boot systems of all those millions of Sun / Apple / IBM systems, a close variant ("Postscript") is present inside even more laser printers, and a slightly more distant relative was found for years in Microsoft "QuickBASIC". There are other examples to taunt the easily panicked programmer. Forth is nearly ubiquitous in computer applications, but is seldom announced, kind of like germs.

And as for Latin, there is only one place in the world where it is the accepted language in daily use: Vatican

City in Rome. As Latin gave rise to closely related "Romance languages" such as Italian, French, Spanish, Portuguese, Romanian, Swiss Romansch and whatever else, Forth will certainly continue to spawn related tongues such as those mentioned in this article.

I am sure that Forth will weather these and other setbacks with STOIC aplomb [12].

6 References

1. `http://www.forth.com/Content/History/History3.htm`.

2. `http://theforthsource.com`.

3. Frenger, P., "A Tribute to FIG-Forth", *ACM Sigplan Notices*, Vol.33 No.8, August 1998, pg.28-31.

4. `http://www.forth.org`.

5. `925-284-1632`.

6. Eckert, B., "Firmware Factory", *ACM Sigplan Notices*, Vol.34 No.12, December 1999, pg.30-33.

7. Sun Microsystems, *Writing F-Code Programs*, 1994, Prentice Hall, Englewood Cliffs NJ.

8. Draheim, G, `comp.lang.forth`, `12-5-2000`.

9. Jakeman, C., "Forth in the UK", *ACM Sigplan Notices*, Vol.35 No.12, December 2000, pg.19-21.

10. Frenger, P., "Forth and the Open Terminal Architecture", *ACM Sigplan Notices*, Vol.34 No. 2, February 1999, pg.36-39.

11. Frenger, P., "Close Encounters of the Forth Kind", *ACM Sigplan Notices*, Vol.36 No. 4, April 2001, pg.21-24.

12. `ftp://ftp.simtel.net/pub/simtelnet/win95/chem/stoic25.zip`.

This is a pun; STOIC was an early variant of Forth on CP/M machines.

Paul Frenger is a medical doctor who has been professionally involved with computers since 1976. He has worked as a computer consultant, published over one hundred articles in the bioengineering and computer literature, edited the ACM SIGForth Newsletter for four years and acquired three computer patents along the way. Paul was bitten by the reverse Polish bug in 1981 and has used Forth ever since. Being both a physician and a computer programmer, Paul believes that the term "hacker" is doubly appropriate in his case.

7 Update

Vendors lost since this publication (there may be useful information on web sites):
- New Micros Inc, TX.
- Mountain View Press, CA (Ref 4).
- Ultratechnology Inc, CA (see web).
- Offete Enterprises, CA (see web).

Vendors still in business:
- Forth Inc, CA.
- Microprocessor Engineering Ltd, UK.
- Mosaic Industries, CA (68HC11, 68HC12).
- GreenArrays Inc, NV (144-cpu Forth chips).
- Harris RTX-2010 rad-hard cpu chips (FPGA equivalent at MPE Ltd).
- IEEE 1275 support: Firmworks, CodeGen, OpenBIOS. IEEE 1275 no longer used by Apple, IBM or Sun, but still is used in "One Laptop Per Child" project.

Other resources available:
- Annual EuroForth meeting / Forth standards.
- Taygeta (free Forth compilers library).
- Various Forth Interest Groups (USA, UK, German, Dutch, Russian, Taiwanese; Silicon Valley FIG.
- FIG listings for various CPU processors.
- Forth Webring.
- "Starting Forth" book online, Leo Brodie.
- "ANS Forth: the New Standard" book online, Jack Woehr.
- "Writing FCode Programs" book online, Sun Microsystems (IEEE 1275).
- Chuck Moore "ColorForth".

Last Sigplan Forth Column: May 2008.